Password Secured Sites – Stepping Forward With Keystroke Dynamics

Sérgio Tenreiro de Magalhães[1], Kenneth Revett[2] and Henrique M. D. Santos[1]
[1] *Universidade do Minho*
Department of Information Systems
Campus de Azurem
4800-058 Guimaraes, Portugal
{psmagalhaes, hsantos} @dsi.uminho.pt
[2] *University of Westminster*
Harrow School of Computer Science
London, UK HA1 3TP
revettk@westminster.ac.uk

Abstract

Computer Authentication is a critical component of most computer systems – especially those used in e-Commerce activities over the internet. Global access to information makes security, namely the authentication process, a critical design issue in these systems. In what concerns to authentication, what is required is a reliable, hardware independent and efficient security system. In this paper, we propose an extension to a keystroke dynamics based security system. We provide evidence that completely software based systems can be as effective as expensive and cumbersome hardware based systems. Our system is a behavioral based system that captures the normal typing patterns of a user and uses that information, in addition to standard login/password security to provide a system that is user-friendly and very effective at detecting imposters. The results provide a means of dealing with enhanced security that is growing in demand in web-based applications based on E-Commerce.

1. Introduction

With the increasing number of E-Commerce based organizations adopting a stronger consumer-orientated philosophy, Web services are also becoming more user orientated. For this strategy to be successful, the authentication (process of confirming an alleged identity) or identification (linking a user to an known identity) of the user must be done with accuracy.

The traditional method for authentication in any information system is the pair login/password. But it is known that the quality of a password, measured by its resistance to attacks, depends of its complexity and, therefore, many of the deficiencies of the login/password systems arise from the limitations of the human memory [1]. Many information systems have increased their level of security by the use of biometric technologies, frequently in association with a card that stores and, sometimes, processes the personal data involved in this process. So, now, a user must present something he knows (password), something he owns (the card) and something he has (the biometrical data).

Despite the generalization of the use of password systems in the Web, Web based applications do not include hardware based biometric technologies. This would place an undue burden on the consumers to have the necessary hardware to support such a security based system. What is ideally required is to have a hardware independent based security system that is as secure as hardware based systems, but without the added expense. It wouldn't make sense to limit the sales of a site to users that own, for instance, a fingerprint reader. But the biometric technologies can measure, mainly, physical characteristics or behavioural characteristics and that can be done in a collaborative way (with the knowledge and active collaboration of the user) or in a stealthy way (without the knowledge of the user) [2]. For instance some behavioural biometrics do not require any special hardware. As an example, a voice recognition system which only requires a microphone that is now very common on personal computers, no longer presents an obstacle. But there is another requirement for authentication processes in Web based applications: the algorithm can't be hardware demanding so that the server can authenticate a few thousands of users almost instantly. The algorithm of keystroke dynamics, a biometrics-based technology greatly enhances security by establishing a pattern in the particular way of a user

to type a text on a keyboard. It was first proposed in the paper "an improved statistical keystroke dynamics algorithm" [2] complies with those requirements and, therefore, they represent a chance for Web based services to take a step forward in the direction of higher security in a realistic way. But the level of accuracy is still an issue and later we will present an improvement on that algorithm that results on a highest level of accuracy.

2. Keystroke Dynamics – previous work

As in many other problems, there have been two different approaches to the challenge of finding an algorithm for keystroke dynamics that minimizes the CER: machine-learning and deterministic algorithms.

Among the solutions based on the machine learning we can find the work presented by Chen [3] which achieved a cross-over error rate (CER) less than 1% and a 0% false acceptance rate (FAR). Ord and Furnell [4] also tested this technology, with a 14 people group, to study the viability of applying it to the simple use of PINs (Personal Identification Numbers) typed on a numeric-pad. Unfortunately the results suggest that, for a large-scale use, the technology is not feasible.

Deterministic algorithms have been applied to keystroke dynamics since the late 70's. In 1980 Gaines [5] presented a report of his work to study the typing patterns of seven professional typists. The small number of volunteers and the fact that the algorithm is deducted from their data and not tested in other people later, results on a lower confidence on the FAR and FRR values presented. But the method used to establish a pattern was a breakthrough: a study of the time spent to type the same two letters (digraph), when together in the text. Since then, many algorithms based on Algebra and on Probability and Statistics have been presented. Joyce Gupta presented in 1990 [6] an algorithm to calculate a value that represents the distance between acquired keystroke latency times and correspondent times previously stored. In 1997 Monrose and Rubin use the Euclidean Distance and probabilistic calculations based on the assumption that the latency times for one-digraph exhibits a Normal Distribution [7]. Later, in 2000, they also present an algorithm for identification, based on the similarity models of Bayes [8], and in 2001 they present an algorithm that uses polynomials and vector spaces to generate complex passwords from a simple one, using the keystroke pattern [9].

The algorithms cited are a small example of the many approaches used to find adequate keystroke dynamics algorithms with a convenient CER. Many others could also be referred, all with different evaluation methods, different number of users involved (usually a limited number of users), different number of keystrokes required to enroll the system and different number of repetitive operations required to authenticate and/or identify the user. This diversity in the algorithm parameters and in the evaluation method makes the task of comparing their results a very difficult one. Furthermore, there is, in this subject, no concept of what is a representative data sample. The same algorithm presents different results when tested with different volunteer groups. The only way to compare two algorithms is to test it against the same group.

Envisaging wide scale applications, like web-based applications (where this method is not executable now) one must consider the results only if the test user group's size is considerably large. In this application domain one must remember that the computational effort necessary to execute the algorithm is a critical factor.

Nevertheless, and for the record, according to Peacock [10] regarding keystroke dynamics, they reported FAR values from 0% to over 50%, the FRR varies from more them 25% to less then 1% and the numbers of users involved is usually between ten and one hundred.

3. An lightweight algorithm

The algorithm presented in [2] is a lightweight an enrollment and an authentication stage. The enrolment process, made by the user once on the first use of the service, consists on typing the users usual password, or passphrase, twelve times. The data is stored and the average, the median and the standard deviation of the times for each digraph is calculated and stored along with the average, the median and the standard deviation for the total time spent on each password/passphrase.

The authentication process is, in the user's point of view, equal to any password method. He/she only has to introduce his/her password like he/she usually does. For each keystroke the algorithm will measure the time latency, defined as *TLP*, and compare it with the one stored. The comparison result will be a hit if and only if

$$Lowest(Average; median) * \left(0{,}95 - \frac{SDesviation}{Average}\right) \leq TLP$$

and

$$TLP \leq Higher(Average; median) * \left(1{,}05 + \frac{SDesviation}{Average}\right).$$

This same calculation is repeated for all the password/passphrase keystrokes, and the results are stored in a Boolean array.

Then a sum A is calculated. The not hit values do not contribute to the sum A. The value 1 is added if the previous value is not a hit (or if it is the first value) and a value 1,5 is added if the previous value is a hit. The final value of A will decide if the authentication process succeeded, or not, according to the threshold defined by the system administrator. For instance, if the threshold is set on 70%, users will only be authenticated to the system if the value A obtained from a given attempt is over 70% of the highest possible value, which is given by: $(number\_of\_characters - 1) * 1.5 + 1$. Finally, if and only if, the login attempt is accepted, the oldest values stored for the latencies are substituted by the corresponding values collected in this successful attempt. This last procedure will allow the data stored to evolve with the user.

The accuracy of the biometric systems is measured by the False Acceptance Rate – FAR, that measures the percentage of illegitimate attempts to login that are successful – and by the False Rejection Rate – FRR, that measures the percentage of legitimate user's denied login attempts. The Crossover Error Rate, or Equal Error Rate, is the value of these rates when they are equal, for a determinate threshold (figure 1). The lowest the FAR, more accurate is the technology.

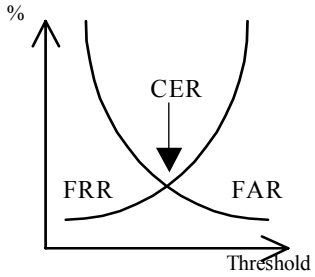

Figure 1 – The Crossover Error Rate (CER) is obtained at the point where FAR=FRR**.**

This algorithm presented a CER of 5,58% and it can achieve, at the lowest thresholds, a FRR of near zero that maximizes the comfort of the user. At the higher demanding thresholds the algorithm presents a near zero FAR, maximizing the security. The several obtainable rates can be found in Figure 2.

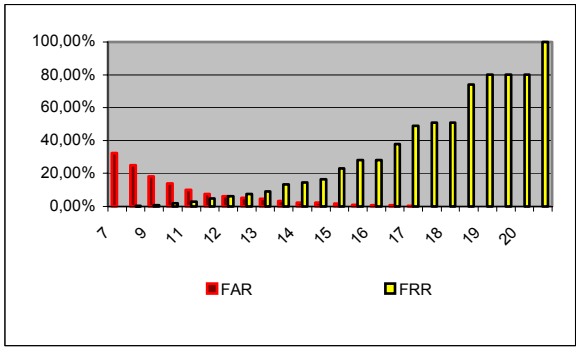

Figure 2 – False Acceptance Rates and False Rejection Rates for the several possible thresholds

4. The performance factors

Revett and Khan [11] concluded that adding keyboard partitioning reduces the impostor success rate (FAR). According to those results, access codes that contained letters from each of the partitions of Figure 3, in a manner that forced the user to enter characters that were scattered across the keyboard, provide more accuracy to keystroke dynamics systems. Furthermore, the speed at which we enter our access codes may actually compromise our access code protection mechanism. According to those results, the best results are achieved when the user does not type at his maximum typing speed.

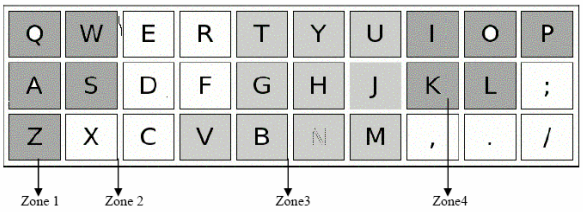

Figure 3 – The keyboard partition into divided four disjoint zones proposed by Revett and Khan [11]

But to impose a password or a typing speed to the user decreases the comfort level and in Web-based applications to lose 1% of the users can mean loosing some thousands. The large-scale factor increases the importance of keeping the system comfortable, non-intrusive and with a low FRR. Therefore the keystroke dynamics algorithms must find a way to include user's performance factors in the decision process, instead of forcing users to change their everyday patterns.

5. A new algorithm that scores the time latencies

The need to include in the decision process the user's performance factors (the same factors for everyone but maybe with different values for the corresponding variables used) generates the need to score the time latencies. We now propose to do that by multiplying the vector coordinates used to generate the sum A in the algorithm described in section four by several variables that must be adjusted for each user if we intend to maximize the accuracy. Unfortunately, we can not assume that the Web users are advanced technology users and, therefore, in this context we must instantiate the variables and find constants, *k*, that will maximize not each user's CER but the general Web community's CER. So, now $A = \sum_{i=1}^{n} \left(k_1 k_2 ... k_p f(t_n)\right)$, where *n* is the number of time latencies resultant from the password string, *p* is the number of performance factors included in the decision process and $f(t_n)$ is the result of applying the previously described algorithm to the n[th] time latency.

The work presented in the paper "Enhancing login security using keystroke hardening and keyboard gridding" [11] suggested that a good factor to start with could be the average typing speed and the empirical data showed that it's weighing factor is $k_{speed} = 2$. Intuition says that users that are right-handed would be more systematic when changing from the left hand to the right hand and the cited work [11] indicated that the characters in contiguous areas (Figure 3) tend to have more stable time latencies. So we decided to give more credit to time latencies that resulted from typing two characters in contiguous zones going from left to right. Also in this case the best constant has the value 2 (two). The result, as showed in the next section, was a reduction on the CER.

6. Evaluation of the algorithm

Establishing the error rates of a biometric technology is a complex issue. Studies have been made to normalize that evaluation, but the results are strongly oriented to physical biometrics and are still strongly dependent on the number of individuals involved in the process and, what is worse, of their characteristics. This means that, even with a large amount of data collected, the results can be very different if we change the group evaluated. This happens because it is very difficult to obtain a sample representative of the population, since we do not know how to characterize the population [2].

The used data was collected through a java applet installed on a web server and on personal laptops. In physical biometrics the user provides a given physical characteristic in different positions and/or conditions, but always the same characteristic. In behavioral biometrics what is presented each time there is an attempt to logon, is a different behavior, so the FAR and FRR must be calculated considering the number of attempts and not the number of users [2]. Anyway, on a Web environment there is no way to know if two attempts to login are from one or two users. The data collected resulted from 170.391 attempts to crash 143 patterns of legitimate users (to calculate the FAR) and from 251 legitimate logins (to calculate the FRR).

Both the previous algorithm, presented in [2] and the now proposed one were used with this data to establish error rates. The results showed that this algorithm has False Acceptance Rates that can be set close to zero (Figure 4) by defining higher threshold values (therefore maximizing the security levels) and False Rejection Rates that can also be set close to zero (Figure 5) by defining lower threshold values (therefore maximizing comfort and availability).

Comparing the global results obtained with the new algorithm (Figure 6) with those obtained with the previous one (Figure 7), we can verify that the Crossover Error Rate was reduced from 5,58% to slightly less then 5%. We can also verify that the different thresholds of the new algorithm correspond to different levels of security, while in the previous one some thresholds, like 15.5 and 16 or 17.5 and 18 corresponded to the same False Rejection Rate once some values were not reachable by the calculating process.

Like with the other algorithm, by varying the thresholds a system administrator, or an IDS, can obtain a FAR value near 0%, a FRR value near 0%, or can establish a balance, somewhere between those extreme values, according to the particularly security needs.

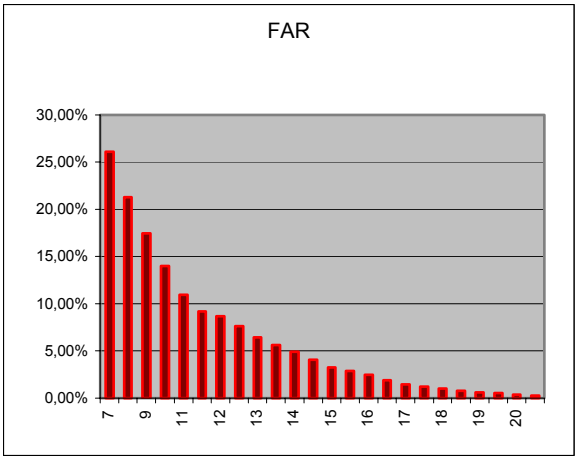

Figure 4 – False Acceptance Rate achieved with the new algorithm for several possible thresholds

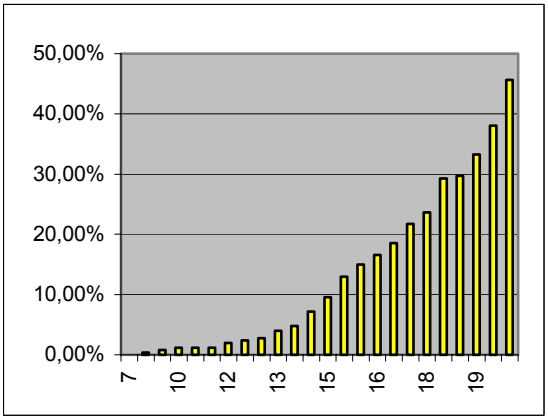

Figure 5 - False Rejection Rate achieved with the new algorithm for several possible thresholds

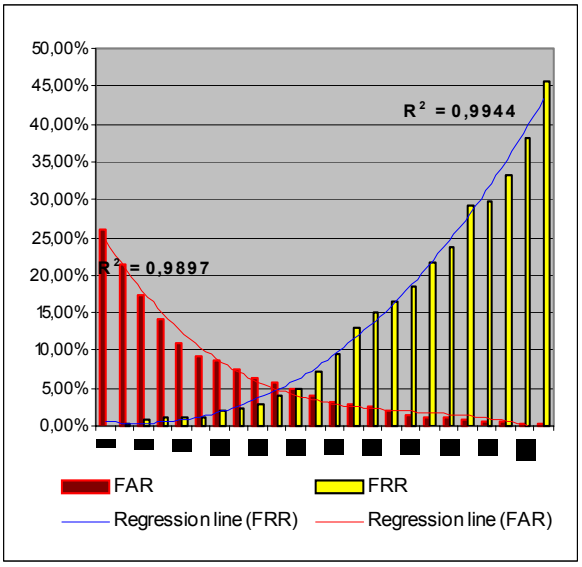

Figure 6 – False Acceptance Rate and False Rejection Rate for several possible thresholds and estimation of the Crossover Error Rate of the new algorithm

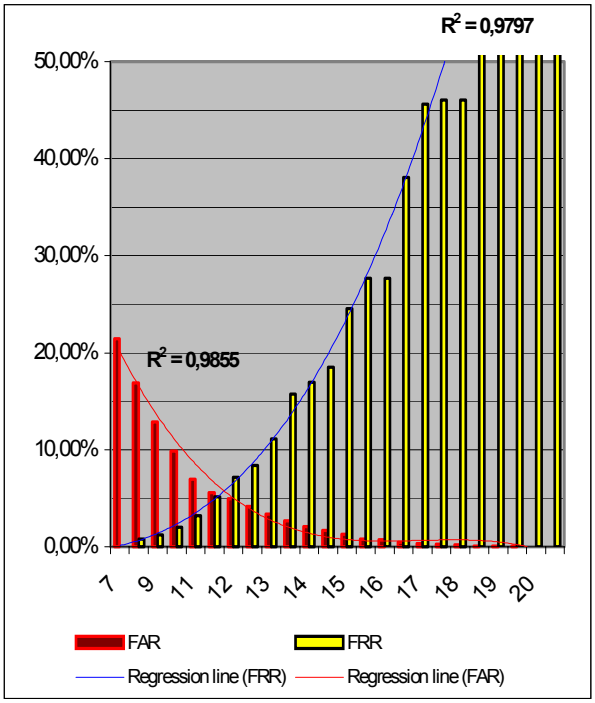

Figure 7 - False Acceptance Rate and False Rejection Rate for several possible thresholds and estimation of the Crossover Error Rate of the previous algorithm when tested with the new data.

7. Conclusions

The results from this study suggests that keystroke dynamics can be an effective means to enhance login security. Our system, based on keystroke dynamics, is not overly burdensome to the user (with some threshold levels it can even be used without his knowledge), very cost-effective, and very efficient in terms of the overhead placed on an Internet based server. We achieve a very low FAR/FRR (both can be placed near to 0%) and a very low CER (less then 5%), compatible with those produced by very expensive hardware based systems. In addition, we have begun investigating additional strategies that can be combined with keystroke hardening, such as keyboard partitioning. Partitioning provides an added layer of security, but requires users to limit their selection of login IDs and passwords. But if security is vitally important to the organisation – such as mission critical e-Commerce sites, then this is a small price to pay to remain in business. A single successful attack can literally put a site into financial bankruptcy. We will explore in addition, the effects of ID/password length and typing speed as additional methods to increase the security level of this system.

Lastly, the system we propose is adaptable in that the stored signatures re automatically updated over time, evolving as users' typing styles evolve over time.

10. References

[1] – Yan, J., Blackwell, A.F., Anderson, R. & Grant, A. , 2004, Password memorability and security: Empirical results, *IEEE Security and Privacy* **2**(5), 25-31.

[2] – Magalhães, S. T. and Santos, H. D., 2005, An improved statistical keystroke dynamics algorithm, *Proceedings of the IADIS MCCSIS 2005*.

[3] – Chen, Z., 2000. *Java Card Technology for Smart Cards*. Addison Wesley, U.S.A.

[4] – Ord, T. and Furnell, S. M., 2000. User authentication for keypad-based devices using keystroke analysis. *Proceedings of the Second International Network Conference – INC 2000*. Plymouth, U.K.

[5] – Gaines, R. et al, 1980. Authentication by keystroke timing: Some preliminary results. Rand Report R-256-NSF. Rand

[6] - Joyce, R. and Gupta, G., 1990. Identity authorization based on keystroke latencies. *Communications of the ACM*. Vol. 33(2), pp 168-176.

[7] – Monrose, F. et al, 2001. Password Hardening based on Keystroke Dynamics. *International Journal of Information Security*.

[8] – Monrose, F. and Rubin, A. D., 1997. Authentication via Keystroke Dynamics. *Proceedings of the Fourth ACM Conference on Computer and Communication Security*. Zurich, Switzerland.

[9] – Monrose, F. and Rubin, A. D., 2000. Keystroke Dynamics as a Biometric for Authentication. *Future Generation Computing Systems (FGCS) Journal: Security on the Web*.

[10] – Peacock, A. et al, 2004. Typing Patterns: A Key to User Identification. *IEEE Security and Privacy*. September/October 2004.

[11] – Revett, K. and Khan, A., 2005, Enhancing login security using keystroke hardening and keyboard gridding, *Proceedings of the IADIS MCCSIS 2005*.